

Short Course on HTML Forms with CGI

1 - Example of minimal HTML Form syntax:

```
<HTML>

<HEAD>

    <TITLE>Teilnehmer Formulare</TITLE>

</HEAD>

<BODY>

    <FORM ACTION=". /test2.mycgi" METHOD="GET">

        <HR>

        <B>Ihre Name: </B>

        <INPUT NAME="Name" TYPE="text" SIZE="53"><BR>

        <B>Ihre Adresse: </B>

        <TEXTAREA NAME="Address" ROWS="6" COLS="53"></TEXTAREA><BR>

        <B>Ihre E-Mail: </B>

        <INPUT NAME="Email" TYPE="Text" SIZE="53"><BR>

        <HR>

        <P><CENTER><INPUT TYPE="submit" VALUE="Senden"></CENTER>

    </FORM>

</BODY>

</HTML>
```

```

#!/bin/sh
echo Content-type: text/html
echo
# This above header and empty echo after it is VERY important.
# Otherwise Apache declare an error..."Content-type" header is needed
# The HTML code enclosed in quotes is also very important.....
echo "<HTML>"
echo "<HEAD>"
echo "<TITLE>This is a CGI test</TITLE>"
echo "</HEAD>"
echo "<BODY>"
echo "<FONT SIZR=4>"
echo "<Center><H1>CGI Environment Variables</H1></Center><BR>"
echo "<HR>"
echo "<FONT SIZE=4>"
echo "<Table border=0>"
  echo "<TR><TD>SERVER_NAME:</TD>      <TD>$SERVER_NAME</TD></TR>"
  echo "<TR><TD>HTTP_HOST:</TD>      <TD>$HTTP_HOST</TD></TR>"
  echo "<TR><TD>HTTP_ACCEPT:</TD>      <TD>$HTTP_ACCEPT</TD></TR>"
  echo "<TR><TD>HTTP_ACCEPT_CHARSET:</TD><TD>$HTTP_ACCEPT_CHARSET</TD></TR>"
  echo "<TR><TD>HTTP_ACCEPT_LANGUAGE:</TD><TD>$HTTP_ACCEPT_LANGUAGE</TD></TR>"
  echo "<TR><TD>HTTP_USER_AGENT:</TD> <TD>$HTTP_USER_AGENT</TD></TR>"
  echo "<TR><TD>HTTP_REFERER:</TD>      <TD>$HTTP_REFERER</TD></TR>"
  echo "<TR><TD>HTTP_CONNECTION:</TD>      <TD>$HTTP_CONNECTION</TD></TR>"
  echo "<TR><TD>SERVER_PORT:</TD>      <TD>$SERVER_PORT</TD></TR>"
  echo "<TR><TD>REMOTE_HOST:</TD>      <TD>$REMOTE_HOST</TD></TR>"
  echo "<TR><TD>REMOTE_PORT:</TD>      <TD>$REMOTE_PORT</TD></TR>"
  echo "<TR><TD>REMOTE_ADDR:</TD>      <TD>$REMOTE_ADDR</TD></TR>"
  echo "<TR><TD>REMOTE_USER:</TD>      <TD>$REMOTE_USER</TD></TR>"
  echo "<TR><TD>SERVER_PROTOCOL:</TD> <TD>$SERVER_PROTOCOL</TD></TR>"
  echo "<TR><TD>REQUEST_METHOD:</TD>      <TD>$REQUEST_METHOD</TD></TR>"
  echo "<TR><TD>REQUEST_URI:</TD>      <TD>$REQUEST_URI</TD></TR>"
  echo "<TR><TD>REMOTE_IDENT:</TD>      <TD>$REMOTE_IDENT</TD></TR>"
  echo "<TR><TD>AUTH_TYPE:</TD>      <TD>$AUTH_TYPE</TD></TR>"
  echo "<TR><TD>CONTENT_TYPE:</TD>      <TD>$CONTENT_TYPE</TD></TR>"
  echo "<TR><TD>CONTENT_LENGTH:</TD>      <TD>$CONTENT_LENGTH</TD></TR>"
  echo "<TR><TD>SCRIPT_NAME:</TD>      <TD>$SCRIPT_NAME</TD></TR>"
  echo "<TR><TD>SCRIPT_FILENAME:</TD> <TD>$SCRIPT_FILENAME</TD></TR>"
  echo "<TR><TD>QUERY_STRING:</TD>      <TD>$QUERY_STRING</TD></TR>"
  echo "<TR><TD>PATH_INFO:</TD>      <TD>$PATH_INFO</TD></TR>"
  echo "<TR><TD>PATH_TRANSLATED:</TD> <TD>$PATH_TRANSLATED</TD></TR>"
echo "</TABLE><BR><HR>"
echo "</FONT>"

--- Display all the CGI Environment Variables list and values -----
echo "<Center><H1>All Environment variables</H1></Center><BR>"
printenv | sort | sed -e 's/.*/$/\&\<BR\>/'
echo "<HR>"

--- Display only CGI Environment Variables created by 'uncgi' -----
echo "<Center><H1>uncgi generated Env. variables</H1></Center><BR>"
printenv | grep "WWW_" | sed -e 's/.*/$/\&\<BR\>/'
echo "<HR>"

----- Change the special codes given by browser for space,
----- newline, @ etc , back in their original -----
# Convert the + to space, %0D%0A to <BR>, %40 to @ -----
ConvertedSTR=`echo "$QUERY_STRING" | sed -e 's/^\%0A/\<BR\>/g' \
-e 's/^\%0D//g' -e 's/^\%40/@\g' -e 's/^\+/\ /g'` \
-----Separate the 3 NAME=DATA -----
param1=`echo "$ConvertedSTR" | cut -d "&" -f 1 `
param2=`echo "$ConvertedSTR" | cut -d "&" -f 2 `
param3=`echo "$ConvertedSTR" | cut -d "&" -f 3 `

-----Separate the NAME and the DATA from the NAME=DATA -----
kw1=`echo "$param1" | cut -d "=" -f 1`
```

```

val1=`echo "$param1" | cut -d "=" -f 2`
kw2=`echo "$param2" | cut -d "=" -f 1`
val2=`echo "$param2" | cut -d "=" -f 2`

kw3=`echo "$param3" | cut -d "=" -f 1`
val3=`echo "$param3" | cut -d "=" -f 2`

----- Display the CGI Environment Variables list and values -----
echo "<Center><H1>CGI Parameters List</H1></Center><BR>"
echo "Parameter 1 = $kw1<BR>"
echo "Value 1 = $val1<BR>"
echo "Parameter 2 = $kw2<BR>"
echo "Value 2 = $val2<BR>"
echo "Parameter 3 = $kw3<BR>"
echo "Value 3 = $val3<BR>"
echo "<HR>"

----- Adding the Name,Address,e-mail to the visitors file -----
echo "$val1,$val2,$val3-$IPAddr: $REMOTE_ADDR" >> visitors.csv
echo "<Center><H1>Besucher Liste</H1></Center>" 
cat visitors.csv | sed -e 's/.*$/&\<BR\>/' 
echo "<HR>"

# - Display which user and group the CGI is identified in Linux system -
echo "<Center><H1>This CGI is identified as: <BR>"
user=`id -nu`
group=`id -ng`
echo "User = $user <BR>"
echo "Group = $group <BR>"
echo "</H1></Center><BR>"
echo "<HR>"

# -----Display all the system Processes -----
echo "<Center><H1>System Processes</H1></Center><BR>"
ps -ax | sed -e 's/ \ \ PID/\<B\>&/' -e 's/.*COMMAND$/&\<\/B\>/' \
-e 's/.*$/&\<BR\>/' 
echo "<HR>"

----- Display free space of all mounted disks in Linux -----
echo "<Center><H1>Disk Space</H1></Center><BR>" 
df -h | sed -e 's/.*$/&\<BR\>/' 
echo "<HR>"

----- Display the Kernel Processes list -----
echo "<Center><H1>Kernel Process Info</H1></Center><BR>" 
procinfo -a | sed -e 's/.*$/&\<BR\>/' 
echo "<HR>"

----- Display who is logged-in now -----
echo "<Center><H1>Who is logged now</H1></Center><BR>" 
w | sed -e 's/.*$/&\<BR\>/' 
echo "<HR>"

----- Display who were the last 20 logins (incl reboots)-----
echo "<Center><H1>Who were the last 20 logins (incl reboots)</H1></Center><BR>" 
last -20 | sed -e 's/.*$/&\<BR\>/' 
echo "<HR>"

----- END of CGI Script -----
echo "</BODY>" 
echo "</HTML>"
```

3- Example of HTML Search machine via HT://Dig

```

<HTML>
  <HEAD>
    <TITLE>Suche durch </TITLE>
  </HEAD>
  <BODY BGCOLOR="#ffffff">      <HR>
    <Table>
      <TR>
        
      </TR>
      <TR>
        <center>
          <form method="GET" action="/cgi-bin/htsearch">
            <font size=-1>
              <H3>Start eine Suche mit </H3>
              <center>

                <select name=method>
                  <option value="and">Und-Verknuepfung der Worte</option>
                  <option value="or" Selected>
                    Oder-Verknuepfung der Worte</option>
                </select>

                <Select name=config>
                  <option value="bashshell">bashshell.conf</option>
                  <option value="forms">forms.conf</option>
                  <option value="htdigv">htdigv.conf</option>
                  <option value="linuxkurs">linuxkurs.conf</option>
                  <option value="manual">manual.conf</option>
                  <option value="samba">samba.conf</option>
                  <option value="selfhtml">selfhtml.conf</option>
                  <option value="webalizer">webalizer.conf</option>
                </Select>

                , Suchbegriffe:
                <input type="text" size="30" name="words" value="">
                <input type="submit" value="Search">

                <select name="sort">
                  <option value="score" selected>Score
                  <option value="time">Time
                  <option value="title">Title
                  <option value="revscore">Reverse Score
                  <option value="revtime">Reverse Time
                  <option value="revtitle">Reverse Title
                </select>
              </form>
            </center>
          </TR>
        </Table>
      </BODY>
    </HTML>
  
```